

1. What does the Simple `List` Component do?

In the given code List is the Component that was wrapping the `WrappedListComponent` using the memo function, that means `List` Component had the same functionalities as `WrappedListComponent` but optimized one, because of using the `memo()`, will reduce the unnecessary re-rendering of the component.

The `WrappedListComponent` was acting as a Parent Component which means there is a child Component, which was rendering inside the `WrappedListComponent`. `WrappedListComponent` contains a state (created by using useState Hook), useEffect HOOK, and one component named as handleClick, this parent component renders a list of items using the `SingleListItem` child component by passing all the required props are index, isSelected, onClickHandler, text. And this child component was inside a `map()` function (another higher-order function). Every time the `item` gets changed, the SingleListItem Component renders a list.

And if the isSelected is `true` then the color of the item was green else red, If there is any click operation on the list item then the selectedIndex will get updated as index through the handleClick Component.

2. What problems/warnings are there with the code?

- In the `WrappedSingleListItem` component, the `onClick` handler should be a function that calls the `onClickHandler` function with the `index` parameter.
- `selectedIndex` and `setSelectedIndex` are declared and defined incorrectly using the useState Hook inside the `WrappedListComponent`.
- The first value returned by useState is the current state (selectedIndex) and a function to update the state (setSelectedIndex). Both are defined in the opposite way.
- When using a `map()` function and rendering the list of elements, we also should specify the `key` to make the list unique from others. So, we have to add a `key` prop for `<SingleListItem />` which resides inside the `WrapSingleListItem` to help React identify each item uniquely.
- In the `WrappedListComponent` we are passing the `selectedIndex` value which was an integer, to the `isSelected` which accepts only Bool values. Also, PropTypes are mentioned, which will throw an error because of the incompatibility of data type.
- In the given Code, PropTypes was used for the `WrappedListComponent` and checking that prop was an array and the data type of the items inside the array are string. But the problem is that using the wrong function.
- When using a `map()` function on an array prop, that array prop should not be null else it will throw an error like the map() function doesn't read undefined values or array.map() is not a function. So, in `WrappedListComponent.defaultProps` items object was defined as null, And this items object was used inside `WrappedListComponent`.

Optimizes and Modified Code:

```
import React, { useState, useEffect, memo } from 'react';
import PropTypes from 'prop-types';

// Single List Item
const WrappedSingleListItem = ({ index, isSelected, onClickHandler, text, }) => {

  return (
    <li
      style={{ backgroundColor: isSelected ? 'green' : 'red'}}
      onClick={() => onClickHandler(index)} // Error1
    >
      {text}
    </li>
  );
};

WrappedSingleListItem.propTypes = {
  index: PropTypes.number,
  isSelected: PropTypes.bool,
  onClickHandler: PropTypes.func.isRequired,
  text: PropTypes.string.isRequired,
};

const SingleListItem = memo(WrappedSingleListItem);

// List Component
const WrappedListComponent = ({
  items,
}) => {
  const [selectedIndex, setSelectedIndex] = useState(); // Error2

  useEffect(() => {
    setSelectedIndex(null);
  }, [items]);

  const handleClick = index => {
    setSelectedIndex(index);
  };
};
```

```

return (
  <ul style={{ textAlign: 'left' }}>
    {items.map((item, index) => (
      <SingleListItem
        key={index} // Error3
        onClickHandler={() => handleClick(index)}
        text={item.text}
        index={index}
        isSelected= { index === selectedIndex} // Error4
      />
    ))}
  </ul>
)
};

WrappedListComponent.propTypes = {
  items: PropTypes.arrayOf(PropTypes.shape({ //Error4
    text: PropTypes.string.isRequired,
  })),
};

WrappedListComponent.defaultProps = {
  items: [ //Error5
    {text:"Jnaneswar"},
    {text:"Rao"},
    {text:"Nerella"}
  ],
};

const List = memo(WrappedListComponent);

export default List;

```

THANK YOU